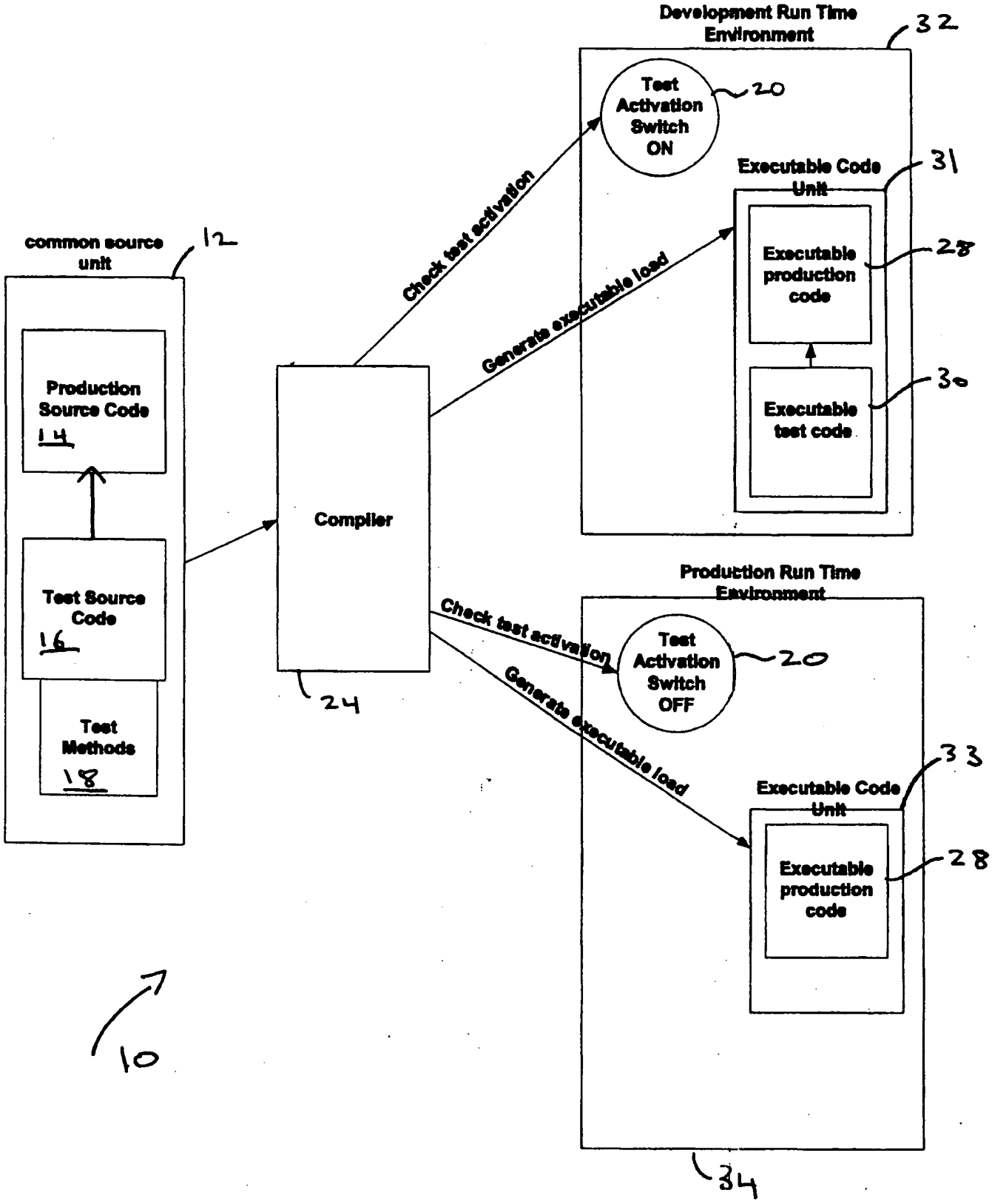

Development Run Time Environment
32
Test Activation Switch ON
20
Executable Code Unit
31
Executable production code
28
Executable test code
30
common source unit
12
Production Source Code
14
Test Source Code
16
Test Methods
18
Compiler
24
Check test activation
Generate executable load
Production Run Time Environment
Test Activation Switch OFF
20
Check test activation
Generate executable load
Executable Code Unit
33
Executable production code
28
10
34

FIG. 1

```
* 1. productive class:
* definition
class OPERATIONS definition.
  public section.
    class-methods:
      ADD importing A type I
                    B type I
          returning VALUE(RESULT) type I. ,
endclass.

* implementation
class OPERATIONS implementation.
  method ADD.
    RESULT = A + B.
  endmethod.
endclass.
```

```
* 2. test class:
* definition
class TEST_OPERATIONS definition for testing.
  public section.
    methods TEST_ADD for testing.
endclass.

* implementation
class TEST_OPERATIONS implementation.

  method TEST_ADD.
*   test data: variable needed to store the result from the productive method:
    data: ACTUAL_RESULT type I.

*   call the method under test:
    ACTUAL_RESULT = OPERATIONS=>ADD( A = 3   B = 5 ).

*   compare the result with the expected value:
    CL_AUNIT_ASSERT=>ASSERT_EQUALS(
      ACT = ACTUAL_RESULT
      EXP = 8
      MSG = 'this is the message which occurs if the test failed'
    ).

  endmethod.
endclass.
```

FIG. 2

4

80

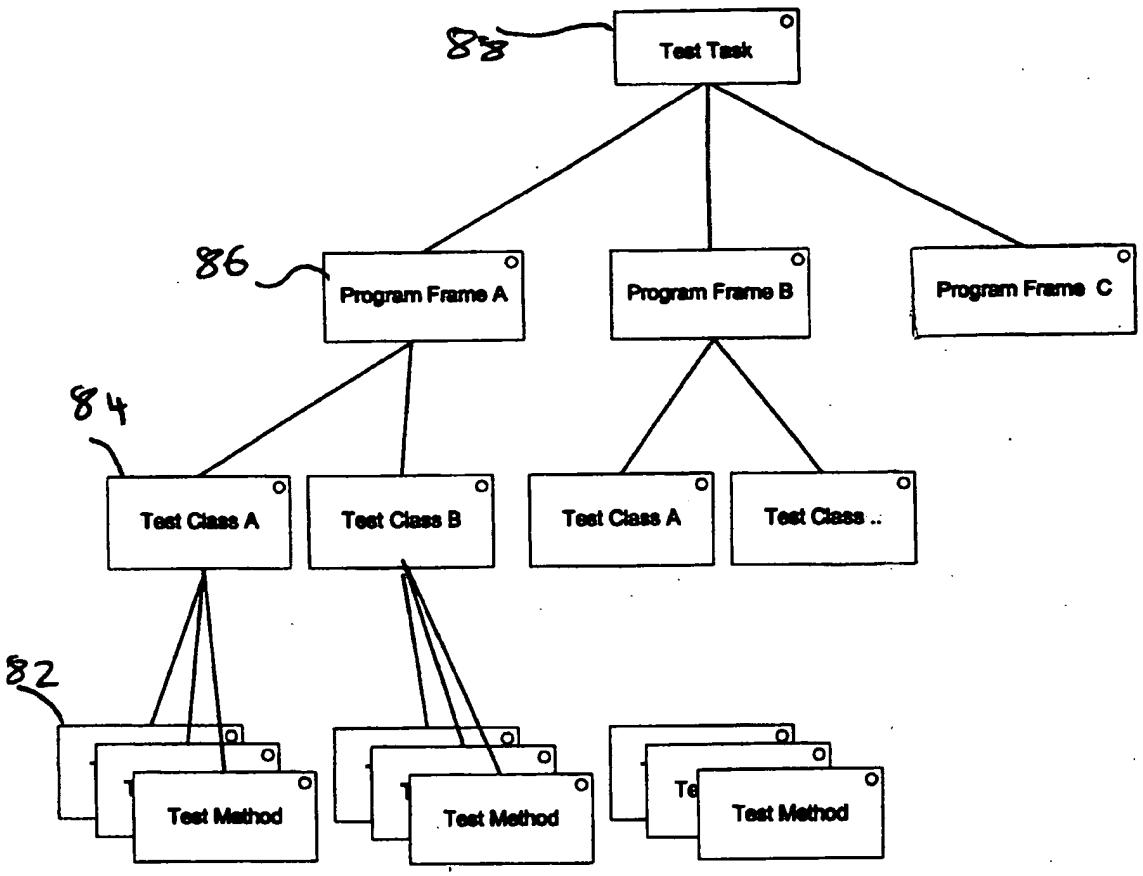
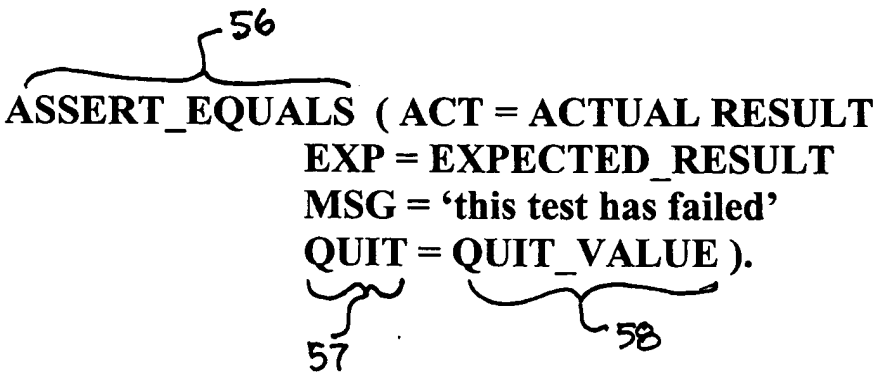

88
Test Task
86
Program Frame A
Program Frame B
Program Frame C
84
Test Class A
Test Class B
Test Class A
Test Class ..
82
Test Method
Test Method
Test Method

FIG. 3




**Where QUIT_VALUE defines at which level the test flow should be interrupted:**

- **NO: continue the current test method.**
- **METHOD: interrupt the current test method.**
- **CLASS: interrupt the test class execution.**
- **PROGRAM: abandon all test class executions of the currently tested program frame.**

# FIG. 4

100

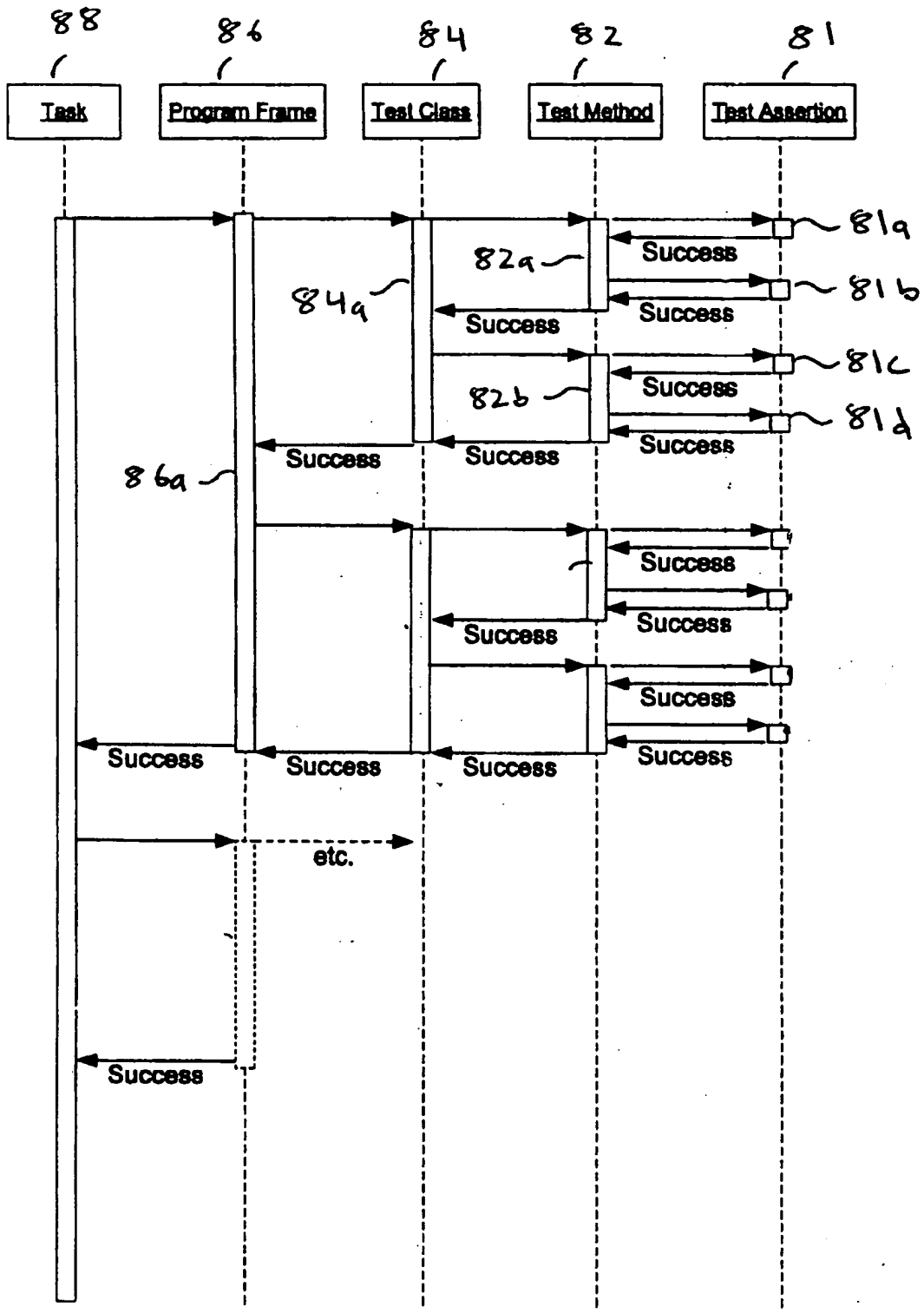

88
86
84
82
81
Task
Program Frame
Test Class
Test Method
Test Assertion
81a
82a
Success
81b
84a
Success
Success
81c
82b
Success
81d
Success
Success
86a
Success
Success
Success
Success
Success
Success
Success
Success
Success
Success
etc.
Success

FIG. 5

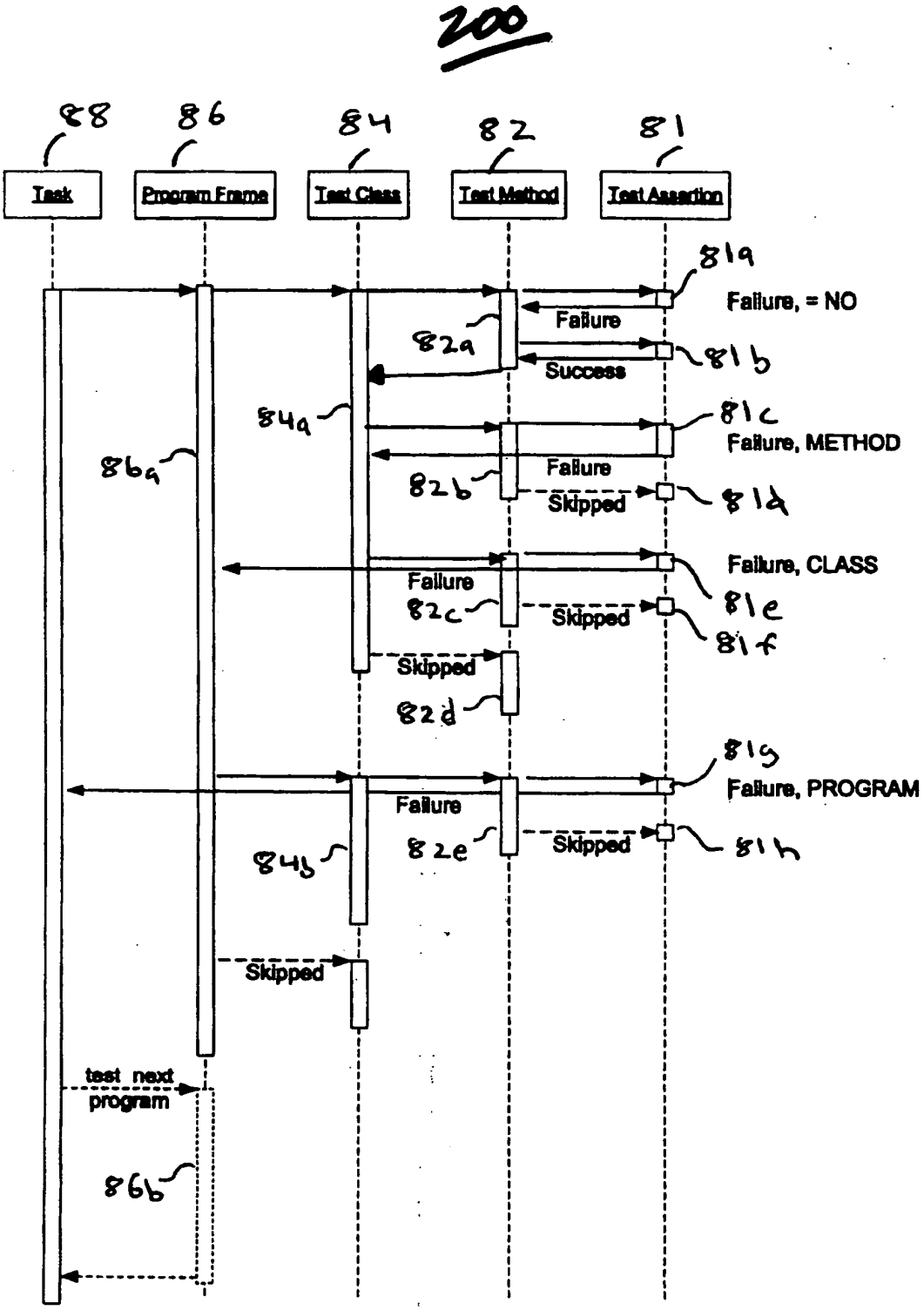

200
88
86
84
82
81
Task
Program Frame
Test Class
Test Method
Test Assertion
81a
Failure, = NO
Failure
82a
81b
Success
84a
81c
Failure, METHOD
86a
Failure
82b
Skipped
81d
Failure, CLASS
Failure
82c
Skipped
81e
81f
Skipped
82d
81g
Failure, PROGRAM
Failure
84b
82e
Skipped
81h
Skipped
test next program
86b

FIG. 6